

Romain Jacquet

BTS SIO 1

24/09/2025

Compte rendu TP Atelier 04 : TD2 CSS : HTML/CSS Avancé (1/2)

Sommaire

Introduction.....	2
Exercice 1.....	2
Exercice 2.....	3
Exercice 3.....	4
Exercice 4.....	5
Exercice 5.....	6
Exercice 6.....	7
Exercice 7.....	8
Exercice 8.....	9
Exercice 9.....	10
Exercice 10.....	10
Exercice 12.....	11
Exercice 13.....	12
Exercice 14.....	13
Conclusion.....	13

Introduction

Dans ce TP, j'ai découvert et appliqué les principales balises et règles de structuration en HTML5 et CSS. L'objectif était de construire progressivement une page web complète et cohérente autour d'un thème précis (la biographie de Chuck Norris), tout en respectant les bonnes pratiques de codage et d'accessibilité.

Chaque exercice m'a permis de comprendre et d'expérimenter une notion précise : la structuration avec `<header>`, `<main>`, `<article>` et `<aside>`, la création de formulaires et de pages liées, la mise en forme avec CSS, l'utilisation des sélecteurs, la construction et le stylage de tableaux, ainsi que la gestion des images et du positionnement. Ce travail m'a aidé à mettre en pratique la logique de séparation contenu/présentation et à consolider mes bases en développement web.

Exercice 1

1 – Mettre la citation dans un `<header>` avec une barre de navigation

J'ai commencé par créer un `<header>` qui contient la citation de Chuck Norris et son auteur, parce que c'est le premier élément qu'on voit en arrivant sur la page. Ensuite, j'ai ajouté une balise `<nav>` à l'intérieur du header, dans laquelle j'ai mis mes liens pour naviguer sur le site (Accueil et Contact).

Le `<header>`, c'est un peu la "bannière" ou l'en-tête de ma page. Et la `<nav>`, c'est la partie où je propose les liens pour aller d'une page à une autre.

2 – Organiser le contenu avec `<main>`, `<article>` et `<aside>`

Pour la suite, j'ai mis tout le contenu principal dans un `<main>`. À l'intérieur, j'ai placé la biographie dans un `<article>`, parce que c'est le texte principal et autonome de ma page.

Ensuite, j'ai mis les "sites amis" et le "Top 10 des facts" dans un `<aside>`, car ce sont des contenus secondaires, qui viennent compléter sans être au cœur du sujet.

En gros, `<main>` désigne la partie principale de la page, `<article>` est un bloc de contenu complet, et `<aside>` sert pour des infos en marge, comme des encadrés ou des liens supplémentaires.

3 – Ajouter un `<footer>` avec un lien de retour en haut

J'ai ajouté un `<footer>` en bas de page avec un petit lien "Retour au début" pour que l'utilisateur puisse facilement remonter. C'est pratique, surtout sur une page longue.

Le `<footer>`, c'est le pied de page, là où on met souvent les infos complémentaires ou les liens utiles.

4 – Mettre les liens dans la barre de navigation

Dans ma balise `<nav>`, j'ai ajouté une liste avec deux liens : "Accueil" pour revenir à la page principale, et "Contact" pour accéder à la page de contact.

J'ai utilisé une liste `` pour bien organiser les liens, et la balise `<a>` pour créer les liens eux-mêmes.

5 – Valider mon code HTML

Pour finir, j'ai validé mon code sur le site officiel du validateur HTML5. Ça m'a permis de vérifier que tout est correct, sans erreurs, et que ma page respecte bien les normes du web.

Valider, c'est important pour s'assurer que la page s'affiche bien partout, et qu'elle sera accessible.

Exercice 2

Construction de la page contact.html

1-

i. **J'ai repris** les mêmes balises `<header>` et `<footer>` que dans la page index.html pour assurer une cohérence visuelle et structurelle sur tout le site. Cela facilite aussi la gestion du site, car ces sections restent identiques d'une page à l'autre.

ii. **J'ai lié** la même feuille de style CSS que pour index.html, ce qui permet de conserver le même style graphique et l'apparence globale.

iii. **J'ai défini** un titre spécifique avec la balise `<title>`, en mettant "Contact - Biographie de Chuck Norris", pour que la page ait un titre clair et pertinent dans l'onglet du navigateur.

Ajout dans la partie principale `<main>`

2-

- i. **J'ai ajouté** un titre principal `<h1>` avec le texte "Adresse" pour introduire la section de contact clairement.
- ii. **J'ai inséré** une image nommée `contact.jpg` pour illustrer que nous sommes à l'écoute, ce qui rend la page plus attrayante visuellement et engageante pour l'utilisateur.
- iii. **J'ai utilisé** la balise `<address>` pour présenter l'adresse complète, car c'est la balise sémantique adaptée pour ce type d'information. J'y ai placé les coordonnées données :
 - Lycée Dominique Villars
 - Place de Verdun
 - 05000 Gap
 - Email : chuckn@yopmail.com

Validation de la page HTML

3-

- **J'ai vérifié** mon code HTML avec le validateur W3C pour m'assurer qu'il respecte bien les standards du web et qu'il ne comporte aucune erreur. Cela garantit une meilleure compatibilité et accessibilité.

Exercice 3

Pour faire les niveaux 1 à 11 et le 14 sur Flukeout, j'ai surtout bien regardé la structure HTML pour comprendre quel élément je devais cibler.

J'ai utilisé plusieurs types de sélecteurs :

- Les **ID**, comme `#plate`, quand je voulais viser un élément précis et unique.
- Les **classes**, par exemple `.pickup` ou `.cookie`, pour cibler plusieurs éléments avec la même caractéristique.
- Les sélecteurs de balise, comme `div` ou `td`, quand c'était simple.
- Les sélecteurs combinés, notamment `>` pour les enfants directs, ou en enchaînant plusieurs classes (`.class1.class2`).
- Les pseudo-classes comme `:nth-child()` pour choisir un élément selon sa position.

Je faisais souvent ça en testant petit à petit : commencer par un simple sélecteur, voir si ça marche, puis ajouter un truc en plus pour être plus précis. Par exemple, cibler `#plate` puis affiner avec `.pickup` si besoin.

Exercice 4

1. J'ai enlevé le soulignement de tous les liens
Pour cela, j'ai utilisé la propriété CSS `text-decoration: none;` sur la balise `<a>`. Cela permet d'afficher les liens sans soulignement.
2. J'ai changé la couleur des liens visités
J'ai ajouté une règle CSS avec le sélecteur `a:visited` et j'ai défini la couleur en bleu clair avec `color: #0088FF;`.

3. J'ai changé la couleur des liens au survol de la souris

Avec le sélecteur `a:hover`, j'ai appliqué la couleur orange (`color: orange;`) pour que les liens changent de couleur quand on passe la souris dessus.

J'ai intégré ces styles dans une balise `<style>` placée dans l'en-tête (`<head>`) du fichier HTML. Grâce à ces modifications, les liens sont désormais plus lisibles, modernes et interactifs.

Exercice 5

1 : Création de la table

Ce que j'ai fait : J'ai créé la structure HTML de la table avec les 7 colonnes demandées.

Comment j'ai procédé :

J'ai utilisé la balise `<table>` pour créer le tableau

J'ai ajouté un en-tête `<thead>` avec une ligne `<tr>` contenant les 7 colonnes

Chaque colonne est dans une balise `<th>` avec un `` à l'intérieur

J'ai placé le tableau dans une nouvelle section de la balise `<aside>`

2 : Ajout de la classe "skill"

Ce que j'ai fait : J'ai identifié les arts martiaux et leur ai ajouté la classe CSS.

Comment j'ai procédé :

J'ai repéré les 6 colonnes d'arts martiaux (de Karaté à Ju-jitsu)

J'ai ajouté `class="skill"` dans les balises `` de ces colonnes

J'ai laissé la colonne "Acteurs" sans cette classe

3 : Insertion des données

Ce que j'ai fait : J'ai ajouté les 6 lignes d'acteurs avec leurs niveaux.

Comment j'ai procédé :

J'ai créé une section `<tbody>` pour le corps du tableau

J'ai ajouté 6 lignes `<tr>` avec 7 cellules `<td>` chacune

J'ai saisi les données acteur par acteur

J'ai vérifié que chaque ligne avait bien 7 valeurs

4 : Test

Ce que j'ai fait : J'ai rendu le tableau lisible et testé le résultat.

Comment j'ai procédé :

J'ai ajouté du CSS pour les bordures et l'alignement

J'ai centré le texte dans toutes les cellules

J'ai vérifié que le tableau s'affichait correctement

J'ai testé que toutes les données étaient bien présentes

J'ai vérifié avec le validator mon index HTML et c'est bon il n'y a aucune erreur

Exercice 6

1.

J'ai ajouté les balises `<thead>` et `<tbody>` autour des parties appropriées du tableau :

- `<thead>` contient la première ligne du tableau (les en-têtes de colonnes)
- `<tbody>` contient les données du tableau (les 6 lignes d'acteurs)

2.

La structure est maintenant conforme aux bonnes pratiques HTML :

- La ligne d'en-tête est dans `<thead>`
- Les données sont dans `<tbody>`

- Les balises fermantes `</thead>` et `</tbody>` sont correctement placées

Pour tester la conformité, J'ai utilisé le validateur HTML du W3C en copiant ce code sur : <https://validator.w3.org/>

Exercice 7

1 : Couleur de fond pour l'en-tête du tableau

Ce que j'ai fait : J'ai appliqué la couleur `#00AAFF` à l'en-tête du tableau.

Comment j'ai procédé :

J'ai utilisé le sélecteur `thead` pour cibler spécifiquement l'en-tête

J'ai ajouté la propriété `background-color: #00AAFF`

J'ai vérifié que seule la ligne d'en-tête était affectée

J'ai confirmé que la couleur bleue s'appliquait correctement

2 : Couleur violette pour les skills du tableau

Ce que j'ai fait : J'ai donné la couleur `#640051` aux skills dans le tableau uniquement.

Comment j'ai procédé :

J'ai utilisé le sélecteur descendant `table .skill` pour cibler précisément

J'ai appliqué `color: #640051` uniquement aux éléments dans le tableau

J'ai vérifié que les skills dans les paragraphes gardaient leur style original

J'ai confirmé l'isolation des styles

3 : Alternance des couleurs de lignes

Ce que j'ai fait : J'ai créé l'alternance blanc/`#CCC` pour les lignes du tableau.

Comment j'ai procédé :

J'ai utilisé `tbody tr:nth-child(even)` pour les lignes paires (blanc)

J'ai utilisé `tbody tr:nth-child(odd)` pour les lignes impaires (`#CCC`)

J'ai restreint l'application au `tbody` pour préserver l'en-tête

J'ai vérifié que l'alternance fonctionnait correctement sur les 6 lignes

Test et validation

Ce que j'ai fait : J'ai testé l'affichage et validé le code.

Comment j'ai procédé :

J'ai vérifié que l'en-tête restait bleue comme demandé

J'ai confirmé que les modifications étaient purement CSS

J'ai testé l'affichage dans le navigateur

J'ai validé que le HTML n'avait pas été modifié

J'ai utilisé le validateur W3C pour confirmer l'absence d'erreurs

Le validateur HTML confirme que le code est conforme sans aucune erreur.

Exercice 8

1. Création d'une cellule pleine largeur

- J'ai ajouté un tableau contenant une seule cellule (`<td colspan="7">`) afin qu'elle occupe toute la largeur des 7 colonnes du tableau des compétences martiales.

2. Mise en valeur du chiffre 5

- J'ai utilisé la balise `` autour du chiffre **5** pour le mettre en évidence, comme demandé.

3. Mise en forme et centrage

- J'ai centré le texte avec `text-align: center;` directement dans le style de la cellule.

- J'ai également ajouté un léger fond gris clair et un padding pour que la phrase ressorte mieux.

Au final, la phrase :

« Il apparaît que Chuck Norris est toujours au top (niveau 5) dans tous les martiaux. »

apparaît bien dans une cellule large, centrée, et le **5** est mis en avant.

Exercice 9

J'ai inspecté la boîte en utilisant les outils de développement du navigateur et **j'ai** analysé son modèle de boîte CSS.

J'ai trouvé dans l'onglet Style des outils de développement la représentation visuelle du modèle de boîte avec ses quatre composants :

J'ai identifié les valeurs suivantes pour chaque partie du modèle :

- **Content** : 214.438 × 79 pixels (largeur × hauteur)
- **Padding** : 20 pixels sur tous les côtés
- **Border** : 5 pixels d'épaisseur
- **Margin** :
 - 5 pixels en haut
 - 50 pixels à droite
 - 5 pixels en bas
 - 50 pixels à gauche

J'ai observé visuellement comment ces quatre couches se dessinent à l'écran, avec :

- Le contenu au centre
- Le padding créant un espace intérieur autour du contenu
- La bordure entourant le padding
- La marge créant un espacement extérieur

J'ai constaté que la marge est asymétrique, avec des valeurs latérales (50px) plus importantes que les valeurs verticales (5px), ce qui crée un espacement particulier autour de l'élément.

Exercice 10

J'ai ajouté les règles CSS demandées dans la section de style du document HTML.

J'ai ajouté du padding vertical de 10px aux titres de sections h1, h2 et h3.

J'ai ajouté du margin vertical de 30px à tous les paragraphes.

J'ai ajouté du padding horizontal de 5px aux éléments ayant la classe skill dans les tableaux uniquement.

J'ai ajouté une bordure de 1px solid de couleur #CCCCCC à tous les titres h3.

Exercice 12

1. Placement de l'image de Chuck jeune à gauche du texte

J'ai ajouté ces lignes de code CSS dans la balise style :

```
#biographie h2:first-of-type + img {  
float: left;  
margin-right: 20px;  
margin-bottom: 10px;  
max-width: 300px;  
}
```

2. Placement de l'image "Beware" à droite du texte

J'ai ajouté ces lignes de code CSS :

```
#biographie h2:nth-of-type(4) + img {  
float: right;  
margin-left: 20px;  
margin-bottom: 10px;  
max-width: 300px;  
}
```

3. Amélioration pour le comportement des paragraphes

J'ai également ajouté cette ligne :

```
#biographie p {  
  overflow: auto;  
}
```

Explication des sélecteurs utilisés :

Pour l'image de Chuck jeune :

- #biographie : limite à la section biographie
- h2:first-of-type : cible le premier h2 ("L'enfance")
 - img : sélectionne l'image qui suit directement ce h2

Pour l'image Beware :

- h2:nth-of-type(4) : cible le quatrième h2 ("Le cinéma")
 - img : sélectionne l'image qui suit ce h2

Exercice 13

Dans la section « *L'enfance* », j'ai divisé le texte en deux paragraphes :

- Le premier commence par « *Carlos Ray Norris, plus connu en tant que Chuck Norris, est un acteur américain né le 10 mars 1940 à Ryan (Oklahoma).* »
- Le deuxième commence par « *Spécialiste en arts martiaux...* », comme demandé.

J'ai créé une règle CSS avec la classe `.clear-left` contenant la propriété `clear: left`.

J'ai appliqué cette classe au deuxième paragraphe en ajoutant `class="clear-left"`.

J'ai conservé le texte original sans changer son contenu.

- Je n'ai pas retenu les modifications « *N'ayant pas l'Oklahoma* » et « *sa amie* », car elles semblaient être des erreurs.

J'ai obtenu comme résultat final :

- Deux paragraphes distincts dans la section « *L'enfance* »
- Une image placée à gauche du premier paragraphe
- Un deuxième paragraphe qui contourne correctement le flottant gauche grâce à `clear: left`
- Une structure conforme aux consignes de l'exercice 13
- Un contenu cohérent avec le reste du document

Exercice 14

Test de position relative sur l'image *chuck-jeune.jpg*

J'ai ajouté dans la section CSS des propriétés de positionnement relatives, initialement commentées, afin de pouvoir les tester temporairement. Une fois décommentées, ces propriétés déplacent l'image de Chuck Norris jeune de 20 px vers le bas et 20 px vers la droite par rapport à sa position normale dans le flux du document.

Ajout des icônes de réseaux sociaux

J'ai implémenté deux versions pour les icônes :

- **Version principale (active)** : j'ai utilisé le positionnement fixe pour que les icônes restent visibles en bas à droite de la fenêtre, même pendant le défilement.
- **Version alternative (commentée)** : j'ai prévu une version avec un positionnement absolu pour placer les icônes en bas à droite du document plutôt que de la fenêtre.

Structure HTML ajoutée

J'ai inséré, juste après l'ouverture de la balise `<body>`, un conteneur regroupant les icônes Facebook et Twitter.

Conclusion

Ce TP m'a permis de mieux comprendre la logique du HTML5 et du CSS, en passant de la simple structure d'une page jusqu'à des fonctionnalités plus avancées comme le positionnement, l'utilisation de pseudo-classes, ou encore la gestion des tableaux. J'ai également appris l'importance de valider régulièrement mon code pour garantir sa conformité aux standards du web et assurer une meilleure compatibilité. Au final, j'ai obtenu un site clair, organisé et fonctionnel, qui respecte les principes fondamentaux de la sémantique et de la présentation en HTML/CSS. Cette expérience m'a donné une méthodologie et des réflexes utiles pour mes futurs projets web.